

## Claims

What is claimed is:

- 5           1.       A method in a computer system for performing a transaction that includes multiple requests, the method comprising:
- an application interface of a client computer receiving a request from an application program;
- the application interface associating a transaction identifier with the request,
- 10       wherein the transaction identifier identifies a transaction that the request is associated with;
- the application interface forming a message including the transaction identifier and the request;
- the application interface sending the message to a server;
- 15       the server receiving the message; and
- the server processing the request in a context of the transaction identified by the transaction identifier in the message.
2.       The method as claimed in claim 1, further comprising:
- 20       the application interface associating a sequence indicator to the request, wherein the sequence indicator indicates in what sequence the server should process the request within the context of the transaction; and
- wherein forming the message comprises including the sequence indicator in the message.
- 25       3.       The method as claimed in claim 1, wherein sending the message comprises:
- opening a connection with the server;
- sending the message over the connection;
- 30       determining whether a timeout period has expired; and

closing the connection when the timeout period has expired.

4. The method as claimed in claim 1, further comprising:  
receiving a reply from the server in response to the request;  
5 determining that the reply includes a redirect request that indicates that the request  
should be sent to another server; and  
sending the message to the another server.

5. The method as claimed in claim 1, further comprising:  
10 the server allocating a database connection to the transaction; and  
the server processing the requests that form a part of the transaction over the  
database connection allocated to the transaction.

6. The method as claimed in claim 5, further comprising:  
15 the application interface including a sequence indicator in the message, wherein  
the sequence indicator indicates in what sequence the server should process the request  
within the context of the transaction; and  
wherein processing the requests comprises processing the requests in an order  
indicated by the sequence indicator.

7. A method in a computer system for a client to perform multiple requests in  
the context of a transaction, the method performed by an application interface associated  
with the client, the method comprising:

receiving a request from an application program;  
25 associating a transaction identifier with the request, wherein the transaction  
identifier identifies a transaction that the request is associated with;  
forming a message including the transaction identifier and the request; and  
sending the message to a server.

8. The method as claimed in claim 7, further comprising assigning the

transaction identifier to the transaction if the request is a request to open a new transaction.

9. The method as claimed in claim 7, further comprising:

5 associating a sequence indicator with the request, wherein the sequence indicator indicates in what sequence the server should process the request within the context of the transaction; and

wherein forming the message comprises including the sequence indicator within the message.

10

10. The method as claimed in claim 7, further comprising:

multiplexing together multiple requests that are destined for the server; and sending the multiple requests to the server in a multiplexed format.

15

11. The method as claimed in claim 7, further comprising:

receiving a reply from the server in response to sending the message; and sending the reply to the application program.

20

12. A method in a computer system for a client to redirect a request from a first server to a second server, the method performed by an application interface comprising:

receiving a request from an application program, wherein the application interface is separate from the application program;

formatting the request into a message;

25

sending the message to the first server;

receiving a reply from the first server in response to the message, wherein the reply includes a redirect request that indicates that the second server is a correct destination for the message; and

30

sending the message to the second server without involving the application program.

13. The method as claimed in claim 12, wherein sending the message to the first server comprises:

- opening a connection with the server;
- 5 sending the message over the connection;
- determining whether a timeout period has expired; and
- closing the connection when the timeout period has expired.

14. The method as claimed in claim 12, further comprising the application  
10 interface storing the redirect request so that subsequent similar requests can be initially directed to the second server rather than to the first server.

15. A method in a computer system for a client to send one or more messages to a server, the method comprising:

- 15 opening a connection with the server;
- sending the one or more message over the connection;
- determining whether a timeout period has expired; and
- closing the connection when the timeout period has expired.

20 16. The method as claimed in claim 15, further comprising:  
initializing and starting a timer when a reply is received from the server; and  
wherein determining whether the timeout period has expired comprises comparing  
the timer value to the timeout period.

25 17. The method as claimed in claim 15, further comprising:  
receiving another request destined for the server within the timeout period; and  
sending the another request to the server over the connection.

30 18. The method as claimed in claim 17, further comprising:  
re-initializing and re-starting the timer when a reply corresponding to the another

request is received from the server.

19. A method in a computer system for a server to manage transactions comprising:

5 receiving a message from a client, wherein the message includes a transaction identifier that indicates that a request specified in the message should be performed in a context of a transaction; and

processing the request in the context of the transaction identified by the transaction identifier.

10

20. The method as claimed in claim 19, wherein the message includes a sequence indicator which indicates a sequence, within the transaction, that the request should be processed, and wherein processing the request comprises processing the request in the sequence.

15

21. The method as claimed in claim 20, further comprising:

sequencing the request by placing the request, and other requests that comprise the transaction, in a transaction queue associated with the transaction; and

20 wherein processing the request comprises processing the request in a sequence of requests in the transaction queue.

22. The method as claimed in claim 19, further comprising:

reserving a database connection for the transaction between the server and a database corresponding to the transaction.

25

23. The method as claimed in claim 22, further comprising:

determining whether a free connection to the database is available; and

if the free connection is available, reserving the database connection by mapping the transaction identifier to the free connection.

30

24. The method as claimed in claim 23, further comprising:  
if the free connection is not available, opening a new connection to the database;  
and  
reserving the database connection by mapping the transaction identifier to the new  
5 connection.

25. The method as claimed in claim 24, further comprising:  
if the free connection is not available, determining whether a maximum number of  
connections to the database has been reached; and  
10 if the maximum number of connections to the database has been reached, stalling  
the request until the free connection is available.

26. The method as claimed in claim 19, further comprising:  
closing the connection when the transaction is committed or aborted.  
15

27. The method as claimed in claim 19, further comprising:  
deallocating the connection when the transaction is committed or aborted.

28. A computer system for processing requests from application programs, the  
20 computer system comprising:

a processor for executing an application interface which receives requests destined  
for the server from at least one application program, associates transaction identifiers  
with each of the requests that are associated with transactions, and forms messages  
including the transaction identifiers and the requests; and

25 an interface to the server, coupled to the processor through a bus, which sends the  
messages to the server.

29. The computer system as claimed in claim 28, wherein the processor is  
further for associating sequence indicators with the requests, wherein the sequence  
30 indicators indicate the order that the requests within a transaction should be performed by

the server.

30. A computer system for processing requests from application programs, the computer system comprising:

- 5 a processor for executing an application interface which receives a request destined for a first server from an application program, formats the request into a message, sends the message to the first server via a network interface, receives a reply from the first server in response to the message, wherein the reply includes a redirect request that indicates that a second server is a correct destination for the message, and  
10 sends the message to the second server via the network interface without involving the application program; and

the network interface, coupled to the processor through a bus, which sends the message to the first server and the second server.

- 15 31. A computer system for processing requests from application programs, the computer system comprising:

- a processor for executing an application interface which receives requests destined for the server from at least one application program, opens a connection with the server, sends the one or more requests over the connection, determines whether a timeout period  
20 has expired, and closes the connection when the timeout period has expired; and

an interface to the server, coupled to the processor through a bus, which sends the one or more requests to the server.

- 25 32. The computer system as claimed in claim 31, wherein the processor is further for sending one or more additional requests destined for the server over the connection during the timeout period.

33. A computer system for processing requests received in messages from one or more clients, the computer system comprising:

- 30 a processor for executing a transaction manager that receives, via a network

interface, a message from a client, wherein the message includes a transaction identifier that indicates that a request specified in the message should be performed in a context of a transaction, and for processing the request in the context of the transaction identified by the transaction identifier; and

5 the network interface for receiving the message from the client.

34. The computer system as claimed in claim 33, wherein the message also includes a sequence indicator, which indicates a sequence, within the transaction, that the request should be processed, and wherein the transaction manager processes the request  
10 in the sequence.

35. The computer system as claimed in claim 33, wherein the transaction manager also reserves a database connection for the transaction between the server and a database corresponding to the transaction.  
15

36. A computer-readable medium holding computer executable instructions, the computer-readable medium for performing a method in a computer system for a client to perform multiple requests in the context of a transaction, the method performed by an application interface associated with the client, the method comprising:  
20

receiving a request from an application program;  
associating a transaction identifier with the request, wherein the transaction identifier identifies a transaction that the request is associated with;  
forming a message including the transaction identifier and the request; and  
sending the message to a server.  
25

37. A computer-readable medium holding computer executable instructions, the computer-readable medium for performing a method in a computer system for a client to redirect a request from a first server to a second server, the method performed by an application interface comprising:  
30

receiving a request from an application program, wherein the application interface



is separate from the application program;

formatting the request into a message;

sending the message to the first server;

receiving a reply from the first server in response to the message, wherein the

5   reply includes a redirect request that indicates that the second server is a correct  
destination for the message; and

sending the message to the second server without involving the application  
program.

10       38.    A computer-readable medium holding computer executable instructions,  
the computer-readable medium for performing a method in a computer system for a client  
to send one or more messages to a server, the method comprising:

opening a connection with the server;

sending the one or more message over the connection;

15   determining whether a timeout period has expired; and

closing the connection when the timeout period has expired.

20       39.    A computer-readable medium holding computer executable instructions,  
the computer-readable medium for performing a method in a computer system for a  
server to manage transactions comprising:

receiving a message from a client, wherein the message includes a transaction  
identifier that indicates that a request specified in the message should be performed in a  
context of a transaction; and

25   processing the request in the context of the transaction identified by the  
transaction identifier.